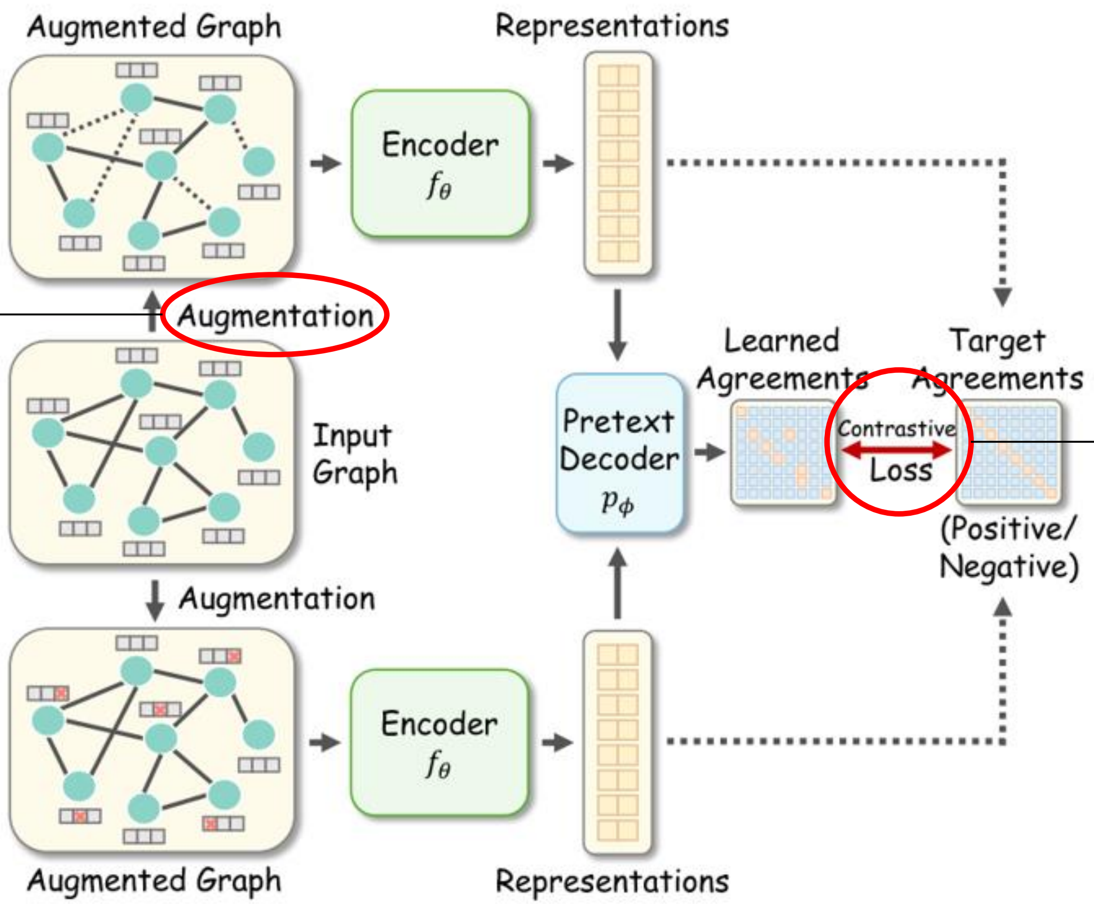
Rethinking and Scaling Up Graph Contrastive Learning: An Extremely Efficient Approach with Group Discrimination

(arxiv, 2022)

paper

Contrastive Learning for GRL

Node Drop,
Edge Drop,
GraphDiffusion ,
Subgraph Sampling …



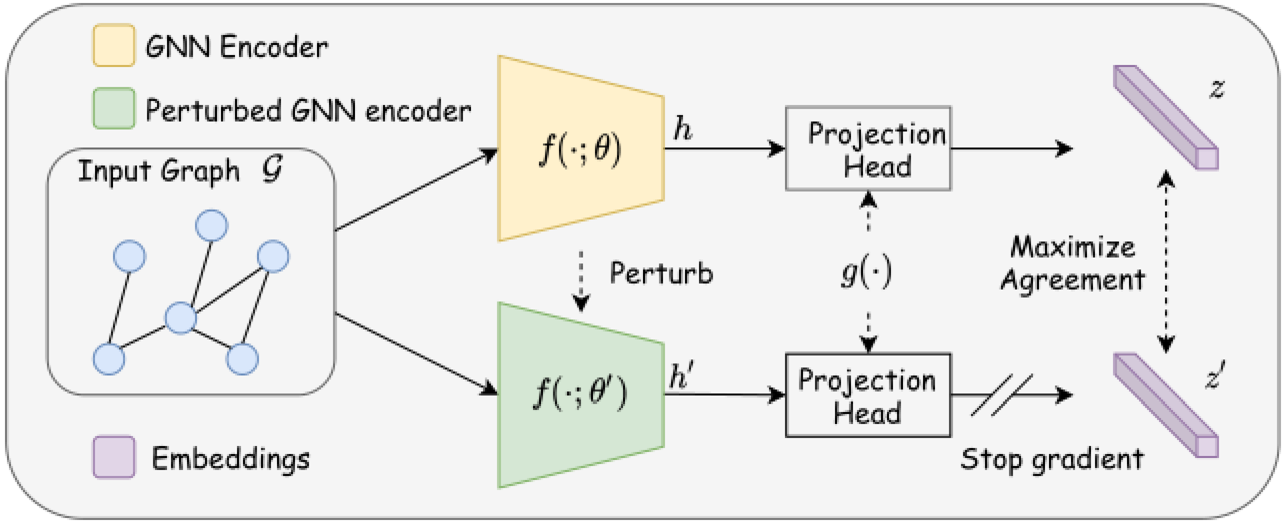
$$L_q = -log \frac{exp(q \cdot k_+/\tau)}{\sum_{i=0}^{k} exp(q \cdot k_i/\tau))}$$

Efficient Contrastive Learning for GRL

◆ Augment aspect

Graph Contrastive Learning with no data augmentation, like SimGRACE, SimGCL

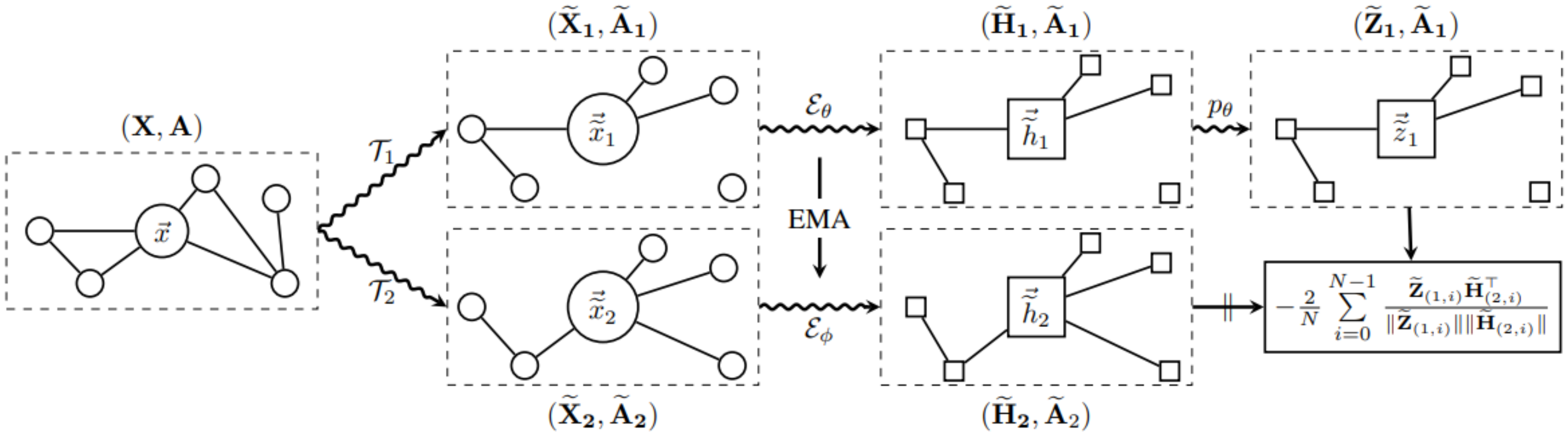
Add random noise to the encoder parameter to construct contrast views



Efficient Contrastive Learning for GRL

◆ **Contrastive Loss/manner**

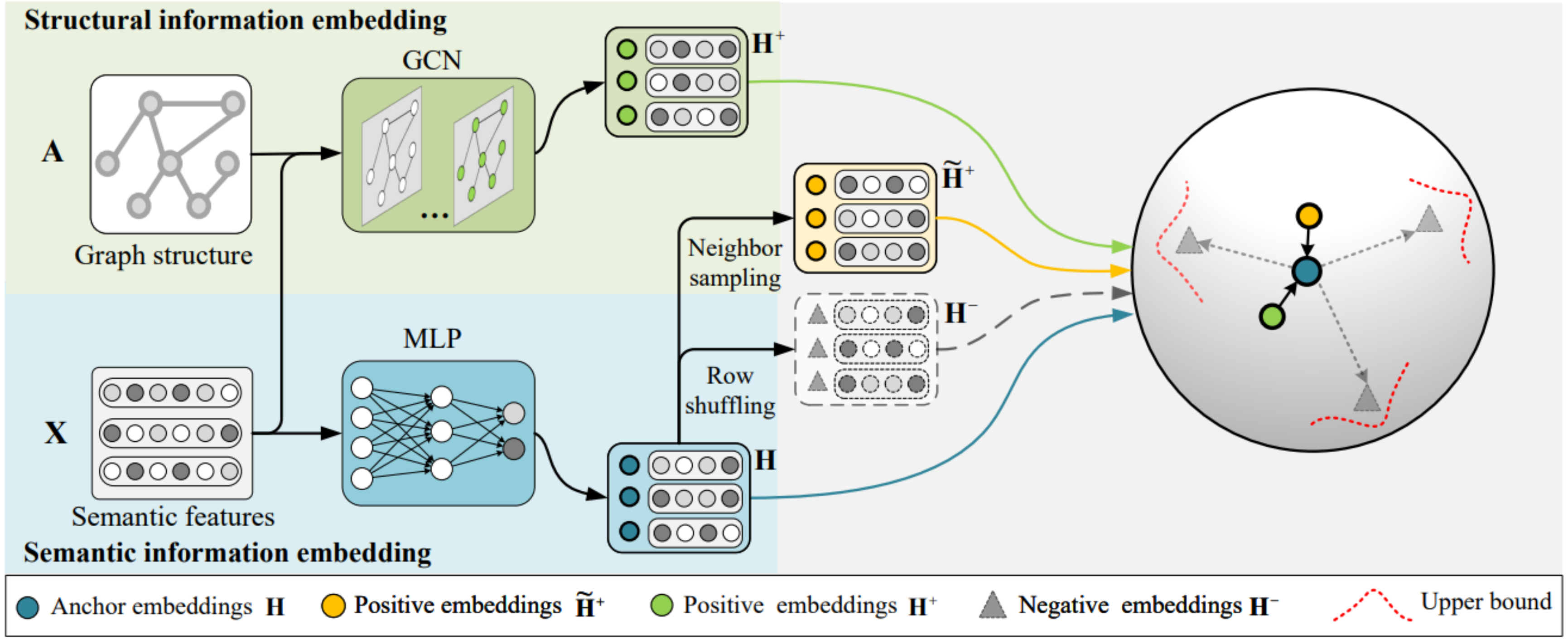
GCL with no negative pairs in loss, like BGRL, GBT



Efficient Contrastive Learning for GRL

◆ Novel Paradigm

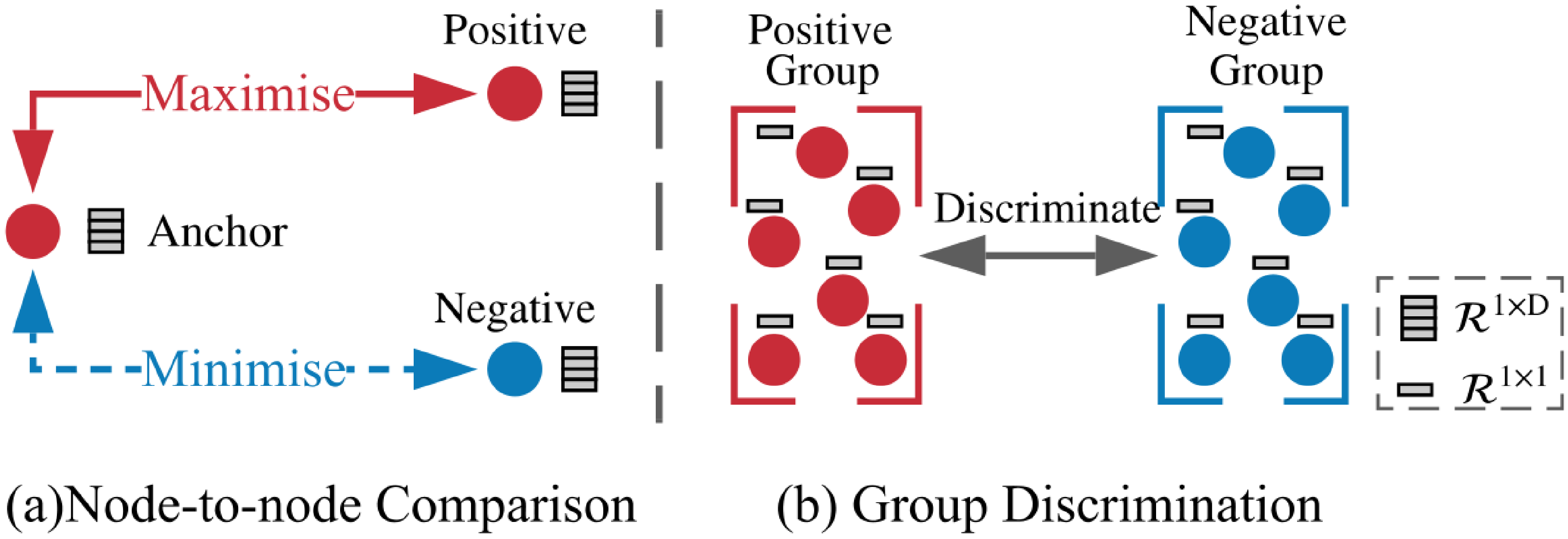
Proposed novel contrastive (self-supervised) learning paradigm, like SUGRL



$$\mathcal{L}_S = \frac{1}{k}\sum_{i=1}^{k}\left\{d\left(\mathbf{h},\mathbf{h}^+\right)^2 - d\left(\mathbf{h},\mathbf{h}_i^-\right)^2 + \alpha\right\}_+ ,$$

$$\mathcal{L}_N = \frac{1}{k}\sum_{j=1}^{k}\left\{d\left(\mathbf{h},\widetilde{\mathbf{h}}^+\right)^2 - d\left(\mathbf{h},\mathbf{h}_j^-\right)^2 + \alpha\right\}_+ .$$

Main Idea



Training time in seconds comparison between GGD and GBT on ogbn-arxiv

Method	Pre	Tr	Epo	Total(E)	Imp(E)	Total(T)	Imp(T)	Acc
GBT(256)	5.52	6.47	300	1,946.52	-	1,941.00	-	70.1
GGD(256)	6.26	0.18	1	6.44	**302.25 ×**	0.18	**10,783.33×**	70.3
GGD(1,500)	6.26	0.95	1	7.21	**269.96×**	0.95	**2,043.16×**	71.6

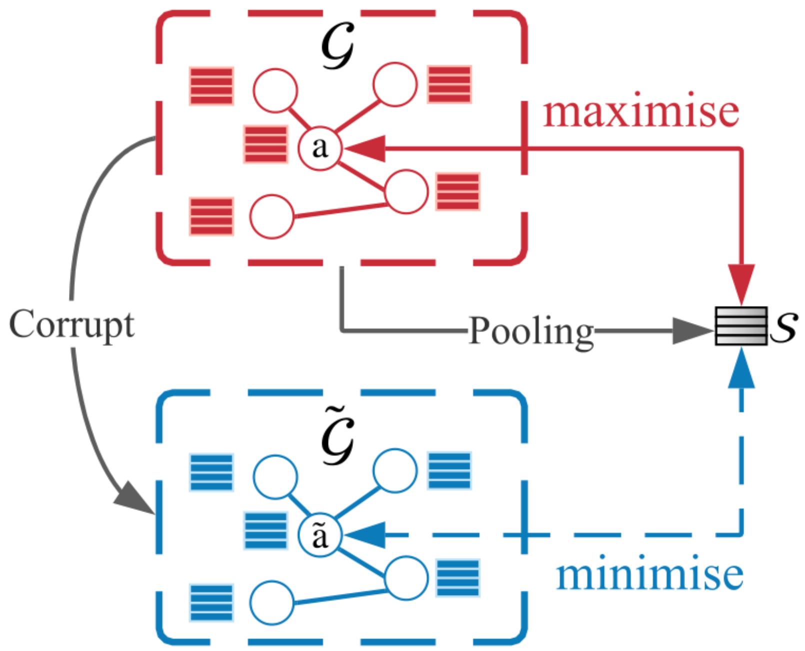
Rethinking DGI

Given graph $\boldsymbol{G}$ with attributes $\boldsymbol{X} \in \mathcal{R}^{N*D}$

Graph $\widetilde{\boldsymbol{G}}$ denotes $\boldsymbol{G}$ with corrupt operate

Obtain node embeddings as $\boldsymbol{z} = GNN(G, X)$,
where $\boldsymbol{z} \in \mathcal{R}^{N*\widehat{D}}$

Obtain summery vector $\boldsymbol{s} = Readout\big(GNN(G, X)\big)$



$$\mathcal{D}(z_i, s) = z_i \cdot w \cdot s$$

$$\mathcal{L}_{DGI} = \frac{1}{2N}\left(\sum_{i=1}^{N} \log \mathcal{D}(z_i, s) + \log(1 - \mathcal{D}(\tilde{z}_i, s))\right)$$

Constant Summary Vector

Observation: the summary vectors are essentially a constant vector ϵI, where ϵ is a scalar and $I \in \mathcal{R}^D$ is an all-ones vector.

Table 2: Summary vector statistics on three datasets.

Statistics	Cora	CiteSeer	PubMed
Mean	0.6225	0.6225	0.6225
Std	5.41e-05	2.86e-05	6.58e-05
Range	0.0036	0.0030	0.0032

- **Xavier initialization**

  The GNN encoder is initialised with **Xavier initialisation** using a uniform distribution, so that the value range of embeddings generated with such an encoder is very small

- **Sigmoid function**

  Sigmoid function is inappropriately applied on the summary vector, which makes the value difference smaller.

The summary vectors contains no useful information.

Constant Summary Vector

Replace the summery vector with different constant vector. Except for 0, the model performance is trivially affected by the value assigned to the constant summary vector.

Table 3: The experiment result on three datasets with changing value from 0 to 1.0 for the summary vector.

Dataset	0	0.2	0.4	0.6	0.8	1.0
Cora	70.3±0.7	82.4±0.2	82.3±0.3	82.5±0.4	82.3±0.3	82.5±0.1
CiteSeer	61.8±0.8	71.7±0.6	71.9±0.7	71.6±0.9	71.7±1.0	71.6±0.8
PubMed	68.3±1.5	77.8±0.5	77.9±0.8	77.7±0.9	77.4±1.1	77.2±0.9

Maybe No need for summary vector as anchor & What truly leads to the success of DGI?

Simplifying DGI & "Group Discrimination"

Simplifying: Predigest the loss proposed in DGI by using an all-ones vector as the summary vector and and simplifying the discriminator.

$$\begin{aligned}\mathcal{L}_{DGI} &= \frac{1}{2N}\left(\sum_{i=1}^{N} \log \mathcal{D}(z_i, s) + \log(1 - \mathcal{D}(\tilde{z}_i, s))\right),\\ &= \frac{1}{2N}\left(\sum_{i=1}^{N} \log(z_i \cdot s) + \log(1 - \tilde{z}_i \cdot s)\right),\\ &= \frac{1}{2N}\left(\sum_{i=1}^{N} \log(\mathrm{sum}(z_i)) + \log(1 - \mathrm{sum}(\tilde{z}_i))\right),\end{aligned} \tag{1}$$

$$\mathcal{L}_{BCE} = \frac{1}{2N}\left(\sum_{i=1}^{2N} y_i \log h_i + (1 - y_i)\log(1 - h_i)\right), \quad h_i = \mathrm{sum}(z_i) \tag{3}$$

Table 4: The experiment result on three datasets with different aggregation function on node embeddings.

Method	Cora	CiteSeer	PubMed
Sum	82.5 ±0.2	71.7 ±0.6	77.7 ±0.5
Mean	81.8 ±0.5	71.8 ±1.1	76.5 ±1.2
Min	80.4 ±1.3	61.7 ±1.8	70.1 ±1.9
Max	71.4 ±1.2	65.3 ±1.4	70.2 ±2.8
linear	82.2 ±0.4	72.1 ±0.7	77.9 ±0.5

Complexity

- InfoNCE

$$\mathcal{L}_{\mathrm{NCE}}(\mathrm{i}) = -\log \frac{e^{z_i \cdot c_i/\tau}}{\sum_{k=1}^{N} e^{z_i \cdot z_k/\tau}}, \qquad O(ND^2)$$

- JSD

$$\mathcal{L}_{\mathrm{JSD}}(\mathrm{i}) = -\log \mathcal{D}(z_i, c_i) + \log(1 - \mathcal{D}(\tilde{z}_i, c_i)), \qquad O(D^2)$$

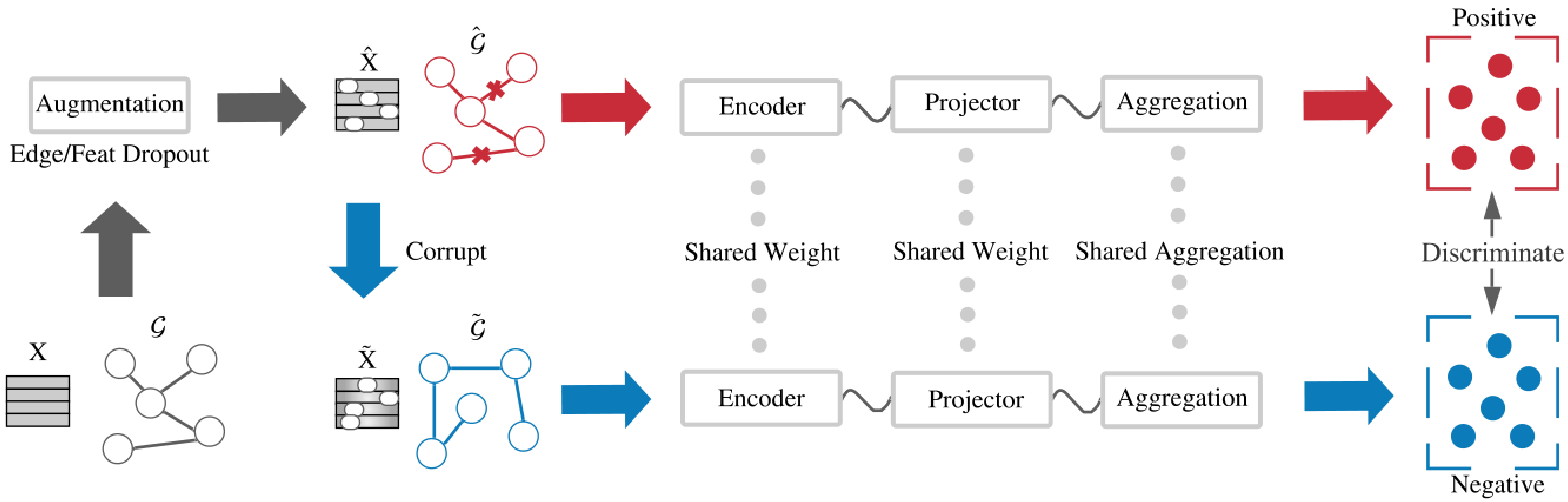
- BGRL

$$\mathcal{L}_{\mathrm{BGRL}}(\mathrm{i}) = -\frac{z_{(\mathcal{G}_1,i)} \cdot h_{(\mathcal{G}_2,i)}}{\| z_{(\mathcal{G}_1,i)} \| \cdot \| h_{(\mathcal{G}_2,i)} \|}, \qquad O(D^2)$$

- GD

$$\mathcal{L}_{\mathrm{BCE}} = \frac{1}{2N}\left(\sum_{i=1}^{2N} y_i \log h_i + (1 - y_i)\log(1 - h_i)\right), \quad h_i = \mathrm{sum}(z_i) \qquad (3) \qquad O(1)$$

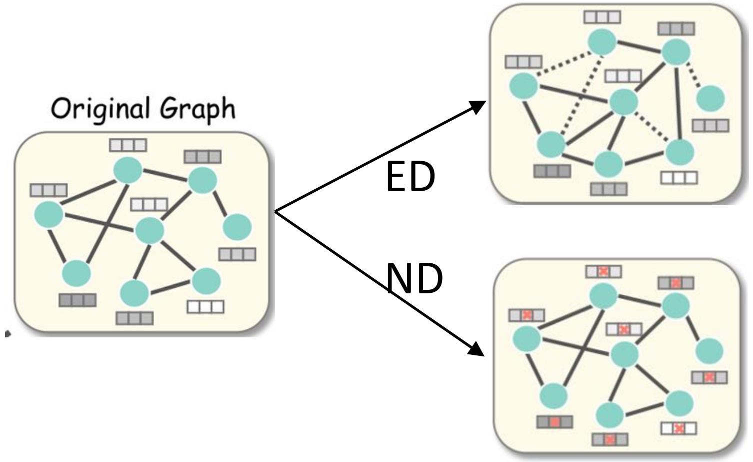
Graph Group Discrimination



Graph Group Discrimination

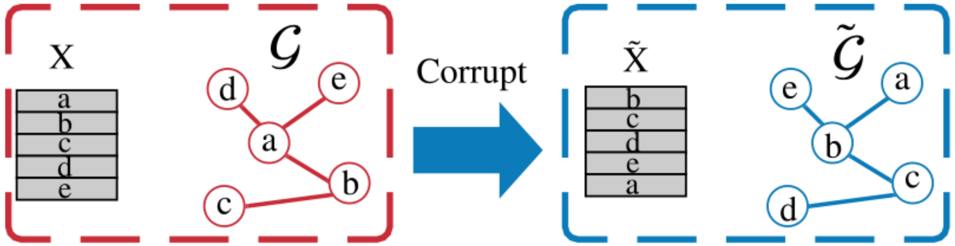
Augmentation

Edge dropout removes a predefined fraction of edges, **node dropout** to mask a predefined proportion of feature dimension.



Corruption

Corrupt original graph as negative group.



Graph Group Discrimination

□ The Siamese GNN

GNN encoder(GCN) + projector head + shared weight.

□ Group Discrimination

Adopts binary cross entropy (BCE) loss to discriminate two groups of node embeddings:

$$\mathcal{L}_{\mathrm{BCE}} = -\frac{1}{2\mathrm{N}}\left(\sum_{\mathrm{i}=1}^{2\mathrm{N}} \mathrm{y_i} \log \mathrm{h_i} + (1-\mathrm{y_i}) \log(1-\mathrm{h_i})\right), \qquad (4)$$

□ Model Inference

$$\mathbf{H}_\theta = \mathbf{GNN}(\mathcal{G}, \mathbf{X})$$

$$\mathbf{H}_\theta^{\mathrm{global}} = \mathbf{A}^{\mathrm{n}} \cdot \mathbf{H}_\theta$$

$$\mathbf{H} = \mathbf{H}_\theta^{\mathrm{global}} + \mathbf{H}_\theta$$

Experiments

Datasets

Table 14: The statistics of eight benchmark datasets.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Amazon Computers	13,752	245,861	767	10
Amazon Photo	7,650	119,081	745	8
ogbn-arxiv	169,343	1,166,243	128	40
ogbn-products	2,449,029	61,859,140	100	47
ogbn-papers-100M	111,059,956	1,615,685,872	100	172

Experiments

Evaluating on Small- and Medium-scale Datasets—Accuracy

Data	Method	Cora	CiteSeer	PubMed	Comp	Photo
X, A, Y	GCN	81.5	70.3	79.0	76.3±0.5	87.3±1.0
X, A, Y	GAT	83.0±0.7	72.5±0.7	79.0±0.3	79.3±1.1	86.2±1.5
X, A, Y	SGC	81.0±0.0	71.9±0.1	78.9±0.0	74.4±0.1	86.4±0.0
X, A, Y	CG3	83.4±0.7	**73.6**±0.8	80.2±0.8	79.9±0.6	89.4±0.5
X, A	DGI	81.7±0.6	71.5±0.7	77.3±0.6	75.9±0.6	83.1±0.5
X, A	GMI	82.7±0.2	73.0±0.3	80.1±0.2	76.8±0.1	85.1±0.1
X, A	MVGRL	82.9±0.7	72.6±0.7	79.4±0.3	79.0±0.6	87.3±0.3
X, A	GRACE	80.0±0.4	71.7±0.6	79.5±1.1	71.8±0.4	81.8±1.0
X, A	BGRL	80.5±1.0	71.0±1.2	79.5±0.6	89.2±0.9	91.2±0.8
X, A	GBT	81.0±0.5	70.8±0.2	79.0±0.1	88.5±1.0	91.1±0.7
X, A	**GGD**	**84.1**±0.4	73.0±0.6	**81.3**±0.8	**90.1**±0.9	**92.5**±0.6

Experiments

Evaluating on Small- and Medium-scale Datasets—Efficiency and Memory Consumption

Time

Method	Cora	CiteSeer	PubMed	Comp	Photo
DGI	0.085	0.134	0.158	0.171	0.059
GMI	0.394	0.497	2.285	1.297	0.637
MVGRL	0.123	0.171	0.488	0.663	0.468
GRACE	0.056	0.092	0.893	0.546	0.203
BGRL	0.085	0.094	0.147	0.337	0.273
GBT	0.073	0.072	0.103	0.492	0.173
GGD	0.010	0.021	0.015	0.016	0.009
Improve	7.3-39.4×	3.4-23.7×	6.9-152.3×	10.7-15.3×	19.2-70.8×

Memory

Method	Cora	CiteSeer	PubMed	Comp	Photo
DGI	4,189	8,199	11,471	7,991	4,946
GMI	4,527	5,467	14,697	10,655	5,219
MVGRL	5,381	5,429	6,619	6,645	6,645
GRACE	1,913	2,043	12,597	8,129	4,881
BGRL	1,627	1,749	2,299	5,069	3,303
GBT	1,651	1,799	2,461	5,037	2,641
GGD	1,475	1,587	1,629	1,787	1,637
Improve	10.7-72.6%	11.8-80.6%	27.2-85.8%	64.5-83.2%	38.0-75.4%

Experiments

Evaluating on Large-scale Datasets

ogbn-arxiv

Method	Valid	Test	Memory	Time	Total
Supervised GCN	73.0±0.2	71.7±0.3	-	-	-
MLP	57.7±0.4	55.5±0.2	-	-	-
Node2vec	71.3±0.1	70.1±0.1	-	-	-
DGI	71.3±0.1	70.3±0.2	-	-	-
GRACE(10k epos)	72.6±0.2	71.5±0.1	-	-	-
BGRL(10k epos)	72.5±0.1	71.6±0.1	OOM (Full-graph)	/	/
GBT(300 epos)	71.0±0.1	70.1±0.2	14,959MB	6.47	1,941.00
GGD(1 epo)	72.7±0.3	71.6±0.5	4,513MB|69.8%	0.18	0.18|10,783×

ogbn-products

Method	Valid	Test	Memory	Time	Total
Supervised GCN	92.0±0.0	75.6±0.2	-	-	
MLP	75.5±0.0	61.1±0.0	-	-	
Node2vec	70.0±0.0	68.8±0.0	-	-	
BGRL (100 epos)	78.1±2.1	64.0±1.6	29,303MB	53m16s	5,326m40s
GBT (100 epos)	85.0±0.1	70.5±0.4	20,419MB	48m38s	4,863m20s
GGD(1 epo)	90.9±0.5	**75.7±0.4**	4,391MB|78.5%	12m46s	12m46s|381×

Experiments

Evaluating on Large-scale Datasets

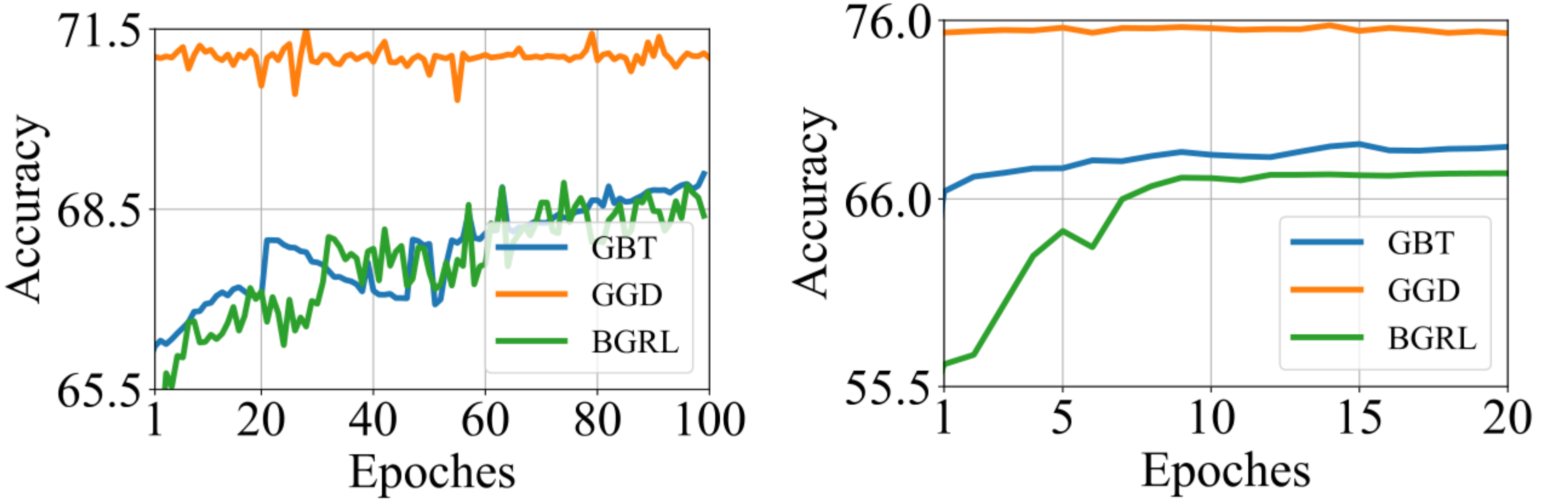
ogbn-papers100M:

Table 12: Node classification result and efficiency comparison on ogbn-papers100M.

Method	Validation	Test	Memory	Time
Supervised SGC	63.3±0.2	66.5±0.2	-	-
MLP	47.2±0.3	49.6±0.3	-	-
Node2vec	55.6±0.0	58.1±0.0	-	-
BGRL (1 epoch)	59.3±0.5	62.1±0.3	14,057MB	26h28m
GBT (1 epoch)	58.9±0.4	61.5±0.5	13,185MB	24h38m
GGD(1 epoch)	60.2±0.3	63.5±0.5	4,105MB|68.9%	9h15m|2.7×

Experiments

Convergence speed comparison



(a) ogbn-arxiv (b) ogbn-products

Table 6: Comparison of DGI using corruption technique and Erdős–Rényi random graphs.

Method	**Cora**	**CiteSeer**	**PubMed**
$\text{DGI}_{\text{corrupt}}$	82.7 ±0.6	71.9 ±0.5	77.9 ±0.7
$\text{DGI}_{\text{Erdos-Renyi}}$	82.6 ±0.4	72.1 ±0.5	79.0 ±1.0

Thanks